MICRO COMMUNICATIONS AND NAVIGATION SYSTEM FOR SHORT-RANGE SPACE AND PLANET-SURFACE COMMUNICATIONS

Martin Agan, Andrew Gray, Edwin Grigorian, David Hansen, Edgar Satorius, Charles Wang

Jet Propulsion Laboratory/California Institute of Technology
4800 Oak Grove Drive
Pasadena, CA 91109-8099

ABSTRACT

This paper provides an overview of the communications system that is being developed as part of the Micro Communications and Avionics Systems (MCAS). The first phase (MCAS1) effort is being focused on a digital binary phase shift key (BPSK) system with both suppressed and residual carrier capabilities. The system is being designed to operate over a wide range of data rates from 1 kbps to 4 Mbps and must accommodate frequency uncertainties up to 10 kHz with navigational Doppler tracking capabilities. As such, the design is highly programmable and incorporates efficient front-end digital decimation architectures to minimize power consumption requirements. The MCAS1 design uses field programmable gate array (FPGA) technology to prototype the real time MCAS1 communications system. Ultimately, this design will migrate to a radiation-hardened, application-specific integrated circuit (ASIC). Specific emphasis in this article is focused on the digital front end and BPSK demodulation portions of the MCAS1 receiver.

1. INTRODUCTION

The objective of the Micro Communications and Avionics Systems (MCAS) effort is to develop chip-level telecommunications systems to meet the unique needs of the National Aeronautics and Space Administration (NASA) short range, low power, space and planet-surface communications. NASA is moving into an era of much smaller space exploration platforms that require low mass and power. This new era is also planning to incorporate in increasing numbers miniature rovers, probes, landers, aerobots, gliders, and multiplatform instruments, all of which have short range communications needs (in this context short range is defined as non-Deep Space Network links). Presently these short range (or *in situ*) communications needs are being met by a combination of modified commercial solutions (e.g., Sojourner) and mission specific designs (e.g., Mars '96, Mars '98, ST-2). The problem with commercial based solutions is that they are high power, high mass, single application-oriented, achieve low levels of integration and are designed for a benign operating environment. The problem with the mission specific designs is that the resultant short-range communication systems do not provide the performance and capabilities to make their use for other missions desirable.

MCAS is primarily targeted at potential JPL users in the space exploration arena such as the Mars Exploration Office (which can use this for various microspacecraft short range communication links such as: orbit/lander, orbiter/rover, orbiter/microprobe, orbiter/balloon, orbiter/sample return canister), ST-4 (orbiter to/from lander link), ST-3 (inter spacecraft links), multiple Discovery missions (e.g., balloons, gliders, probes), ST-5 (space/space and space/ground links), nano-adjunct spacecraft and spacecraft "black box" recorders. MCAS also has applicability to any space mission that has a short-range communications requirement such as Space Station Intravehicular and Extravehicular wireless communications links, X-33 wireless sensor and short range ground links. MCAS is a multiphase effort that will evolve over time taking advantage of advances in communications integrated circuit (IC) technology that will lead to increasingly more integrated solutions, with the eventual goal being the inclusion of microelectromechanical systems (MEMS) oscillators and filters onto single chip transceivers.

The primary goal of MCAS1, which is the first prototype being developed under the MCAS effort, is to achieve a higher level of system integration at the chip level thus allowing significant mass, power, and size reductions, at lower cost, for a broad class of very small platforms requiring short range communications.

Towards this end a design approach has been devised that takes advantage of commercial IC advances where they are applicable to the space environment and utilizes custom design where performance and feature requirements dictate. The realization of this approach has resulted in maximizing the transceiver functions performed in the digital domain. These digital functions will initially be implemented with field programmable gate array (FPGA) technology for purposes of real time demonstration and testing. The final MCAS1 design will then incorporate the digital functions into an application specific integrated circuit (ASIC). The resulting single digital ASIC will be fabricated in a radiation hardened process. The transceiver functions that must be in the analog domain consist primarily of the RF up and down conversion. The approach with the RF subsystem design is to use space-qualified parts where available and leverage the large investment that industry has made in developing highly integrated devices for the commercial wireless markets. A space qualified RF design will be developed through proper selection of these parts (i.e., selection of GaAs components for their inherent radiation hardness) and working with the JPL Parts Section to up-screen them.

Emphasis in this paper will be focused primarily on the digital portion of the transceiver including the data modulation process (Section 2), the receiver front-end processing (Section 3) and the demodulation process (Section 4). A complete description of the MCAS1 transceiver design will be provided in an upcoming report[1].

2. MCAS1 DATA ENCODING AND WAVEFORM MODULATION

This section provides a description of the MCAS1 encoding process from the baseband input bits to the binary phase shift key (BPSK) modulator. First however we note that to be compliant with the proposed Consultative Committee for Space Data Systems (CCSDS) proximity link recommendation, a V.35 scrambler/descrambler is incorporated into the MCAS1 transceiver for optional use with uncoded bit transmissions. The use of scrambling helps to ensure that a sufficient density of data transitions occurs in the transmitted data to aid in the bit timing recovery at the receiver.

The next step after scrambling the transmit bit stream is the differential encoding of the bits. Due to the inherent phase ambiguity of the BPSK constellation, differential encoding can be utilized to transmit the difference in phases between consecutive bits rather

than the actual bits themselves, thus obviating the need to determine the absolute phase at the receiver. The processing performed in the transmitter to implement the differential encoding is given by the following relationship:

$$y(k) = x(k) + y(k-1),$$

where $x(k)$ is the input; $y(k)$ is the output and both are logical 0 or 1. Differential encoding may be enabled or disabled.

Following differential encoding is convolutional encoding to provide error detection and correction capability. The convolutional encoder is the optimal, (in terms of free distance) constraint length 7 rate 1/2 code depicted in Figure 1. The inverter is included to make the encoder compatible with the Standard NASA K=7, r=1/2 convolutional code and by association the proposed CCSDS Proximity Links Recommendation. This inverter ensures there are transitions in the symbols when an all zero's bit pattern is input to the encoder. The inverter may be switched out of the circuit if desired. Two symbols are generated for each input bit into the encoder; consequently, the channel symbol rate is twice the input bit rate. The convolutional encoding may be disabled when uncoded operation is desired.
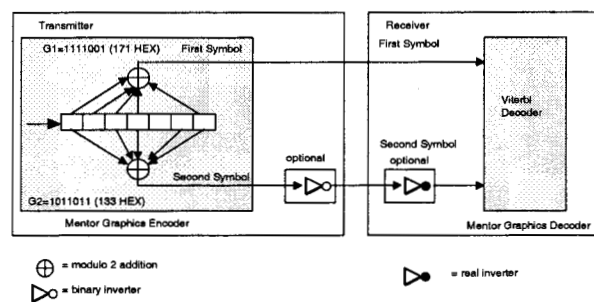

Figure 1. Convolutional Encoder/Decoder (Rate 1/2, 171, 133 Generators).

Normally a squarewave pulse shape is transmitted with an associated non-return-to-zero-level (NRZ-L) waveform, i.e., the binary 1/0 output from the convolutional encoder is routed directly to the phase modulator. When required the transmitter can be set to Manchester encode the transmitted symbols. Manchester encoding (also known as bi-phase-level) represents a binary one as a one for the first half of the bit period and a zero for the second half of the bit period. Manchester encoding a zero translates to a zero during the first half of the bit period and a one for

the second half of the bit period. Because of its spectral shape Manchester encoding will generally be enabled when residual carrier modulation is utilized to prevent the modulated data from interfering with the performance of the receiver carrier tracking and data detection circuits (see Section 4.1).

After Manchester encoding, the encoded baseband data are used to phase-modulate the carrier. The required output from the MCAS1 transceiver is a phase modulated waveform centered at a frequency of 437.1 MHz or 401.585625 MHz that is the input to the diplexer or antenna. This BPSK modulation is achieved through the use of a phase modulator device which will have as inputs a 437.1 MHz or 401.585625 MHz analog carrier and the encoded baseband data. The output is either a 437.1 MHz or 401.585625 MHz phase modulated waveform. The output waveform is geometrically described by the signal constellation as illustrated in Figure 2. This constellation depicts the phase of the output signal when it is translated to baseband. For BPSK modulation, a logical zero is mapped into a phase of zero radians and a logical one is mapped into a phase of π radians. As indicated in the Figure 2, the modulator will have the capability to transmit either supressed carrier or a residual carrier with a 1 radian modulation index.
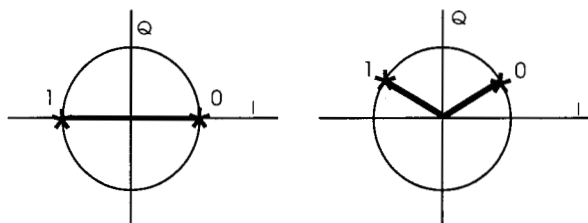

Figure 2. Phase Modulator Transmitted Signal Constellation: Suppressed Carrier and Residual Carrier.

The BPSK modulated transmit signal is amplified by the power amplifier which nominally transmits 500 mW. Following the power amplifier a notch filter centered at the receive frequency is utilized to attenuate transmit spurious signals in the receive band to ensure that the spur power level is well below the receiver sensitivity. The 500 mW power amplifier can also be used to drive a higher power amplifier if required by the operational scenario. The design is specified to accommodate a transmit power amplifier of up to 10 W (i.e, a 10 W power amplifier is baselined that can be driven by 500 mW, and the

allowed receive spur level specification must be met for a 10 W transmit level).

3. MCAS1 RECEIVER FRONT-END PROCESSING

In this section, we describe the MCAS1 receiver front-end. This comprises the automatic gain control (AGC), the analog-to-digital converter (ADC) and the digital downconverter/decimator. These are described separately in this section.

3.1 AGC

The AGC controls the voltage level input to the ADC based on a control voltage signal generated digitally in the FPGA/ASIC (described below). The AGC amplifier provides a 60 dB dynamic range with a typical transfer curve as depicted in Figure 3. As is seen, the gain is approximately linear over the control voltage range from 1.5 to 3.5 volts. For AGC control voltage levels less than 1.5 volts, the AGC gain saturates at 30 dB (weak input signal limit) whereas for control voltage levels above 3.5 volts, the AGC gain limits at approximately −30 dB (strong signal limit). In the latter case, input signal levels from the IF filter which exceed the AGC dynamic range will cause the ADC to saturate thereby creating clipping distortion and thus forcing the ADC to approach the 1 bit performance limit.
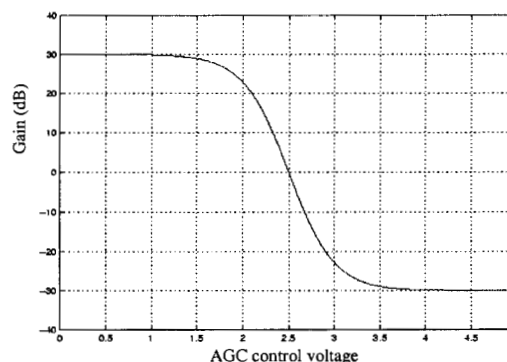

Figure 3. AGC Transfer Curve.

Insofar as the ADC input dynamic range is concerned, the 60 dB AGC dynamic range is more than sufficient. Specifically, the input received signal level can vary over a 70 dB range from −140 dBm to −70 dBm depending on the data rate, transmitter-receiver range, etc. However, because of the wide IF filter bandwidth (6.5 MHz), the corresponding variation in total input power to the ADC is only approximately 30 dB: -103

dBm to −70 dBm, which could readily be maintained by the AGC *if only* the AGC were being controlled to maintain the ADC input power at a fixed level. In actuality, the AGC is being controlled to maintain a constant power level at the output of the Costas arm filters (Section 4), where the bandwidth is generally much narrower than the IF filter bandwidth. This is depicted in Figure 4. The net result is a 30 dB signal dynamic range capability at the lower data rates (1 – 4 kbps) and extending up to 60 dB at the higher data rates (32 kbps and above)[1]. This is more than sufficient to accommodate the above signal dynamic range requirements.
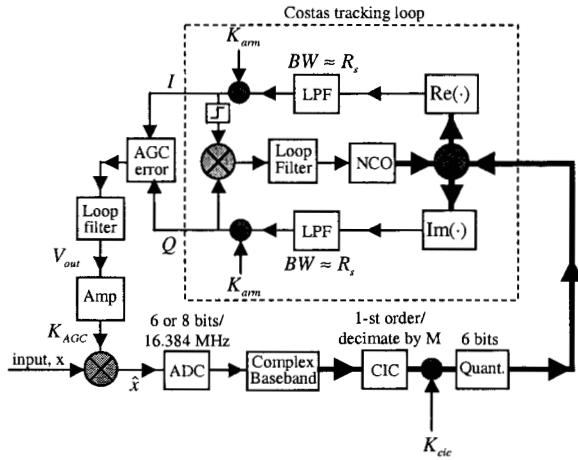

Figure 4. Single-loop AGC design. (The heavy dark lines denote complex data paths.)

3.2 ADC

The MCAS1 receiver employs first-order, bandpass sampling wherein the IF frequency band is mapped directly down to digital baseband. Denoting the IF and ADC sampling frequencies by f_{IF} and F_s, respectively, bandpass sampling at IF imposes the following relationship:

$$f_{IF} = (2n+1) \cdot F_s / 4 ,$$

where n is a positive integer.

In designing the bandpass sampling system, the ADC sampling rate was chosen to accommodate an integral, power-of-two number of samples per symbol at all symbol rates: 4.096 Msys/sec, 2.048 Msys/sec, ..., 1 Ksys/sec. To achieve a minimum of 4 samples per symbol at the highest symbol rate of 4.096 Msys/sec,

an ADC sampling rate of $F_s = 16.384\,MHz$ was chosen. Given F_s, admissible IF frequencies are obtained from the above relationship between f_{IF} and F_s. Based on the availability of IF filters and other considerations, an IF near 70 MHz is desired. The closest admissible IF frequency occurs when $2n+1 = 17$ corresponding to:

$$f_{IF} = 17 \cdot 4.096\,MHz = 69.632\,MHz .$$

To accurately bandpass sample at this IF, the ADC full power bandwidth must be at least 70 MHz.

In addition to the ADC sample rate, the number of bits must be considered. In Figure 5 plots are presented of the ADC quantization noise-to-input power ratio (dB) versus the input scaling or "loading" factor for a 4 bit, 6 bit and 8 bit ADC at different input signal-to-noise power ratios (SNR's). As is seen, there is always an optimal loading point for each size ADC. Above this point (less negative loading factor) clipping distortion limits ADC performance whereas below this point ADC quantization noise is the limiting factor.
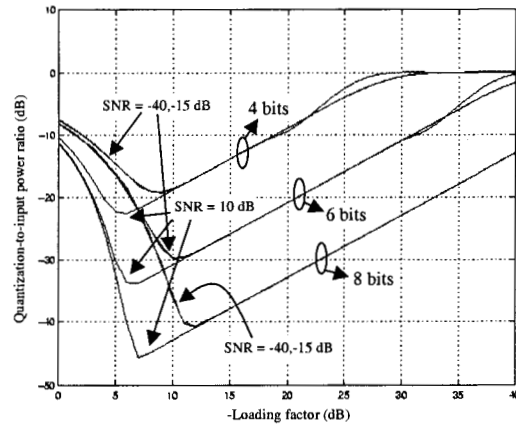

Figure 5. Quantization noise-to-input power ratio.

So for best results with an 8 bit ADC at either −40 dB or −15 dB SNR, the input should be scaled such that its RMS level is approximately −12 dB relative to the ADC full-scale voltage. For a 6 bit ADC at either −40 dB or −15 dB SNR, the optimal loading point is approximately -10 dB relative to full-scale and for a 4 bit ADC at either −40 dB or −15 dB input SNR, it is about -8 dB relative to full-scale. As the input SNR increases to 10 dB, the optimal loading point increases depending upon the number of ADC bits. Based on

dynamic range considerations, quantization noise degradation and size/power constraints[1], an 8 bit ADC has been chosen as the most reasonable choice for MCAS1 implementation.

3.3 DIGITAL DOWNCONVERSION AND DECIMATION

Digital downconversion and decimation directly follow the ADC. A digital complex baseband downconversion scheme was chosen based on considerations of computational efficiency and flexibility. This approach is depicted in Figure 6 and comprises: (1) digital complex mixing from $F_s / 4 = 4.096\ MHz$ down to baseband followed by (2) digital decimation via a first-order, cascaded integrator-comb (CIC) filter[3]. Note that the digital mixing functions do not require multiplication and furthermore the CIC filters are multiplierless and thus the entire structure can be implemented efficiently in the FPGA/ASIC. Also indicated in Figure 6 are the typical data bitwidths used to implement the digital downconverter and decimator (all data are represented in two's complement notation and all indicated data bitwidths include the sign bit).
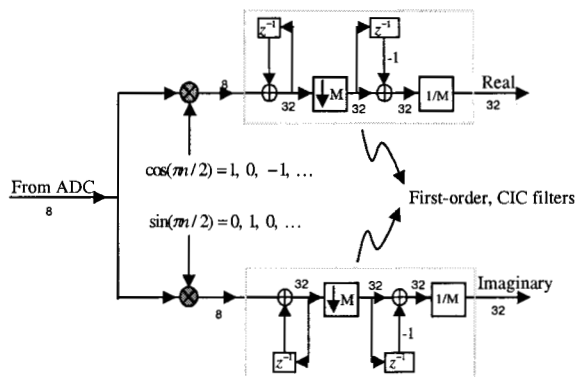

Figure 6. Digital complex basebanding and decimation.

The decimation factor M is programmable and is dependent upon the input data rate. To accommodate symbol timing recovery (Section 4.3), M is typically chosen so that there are at least 16 samples per symbol after decimation, except at the highest data rates. So at 1.024 Msys/sec, 2.048 Msys/sec or 4.096 Msys/sec, M will nominally be set to 1 (no decimation) in which case the remainder of the digital receiver (Costas loop, symbol timing recovery, etc.) will run at the input sampling rate, 16.384 MHz. As the data rate

is lowered below $R_s = 1.024\ Msys / \sec$ down to 8 Ksys/sec, M is increased proportionately such that:

$$\frac{F_s}{R_s \cdot M} = 16.$$

Below $R_s = 8\ Ksys / \sec$, M remains fixed at 128 to accommodate Doppler offsets. Note that as M increases up to 128, more of the input noise to the ADC is filtered out by the CIC filters thereby reducing the total CIC output power. This necessitates a re-scaling operation after the CIC filters.

4. MCAS1 DEMODULATION

In this section, the various elements of the demodulation process are presented including the carrier recovery loop (Section 4.1); the Doppler frequency extraction for navigation (Section 4.2); the symbol timing recovery (Section 4.3); the convolutional decoder (Section 4.4); the differential decoder (Section 4.5) and the descrambler (Section 4.6).

4.1 CARRIER TRACKING LOOP

The carrier tracking loop portion of the MCAS1 transceiver is designed to acquire and track the phase of the received signal. The signal can be suppressed-carrier BPSK, residual-carrier BPSK with modulation index of 1 radian, or unmodulated. The carrier tracking loop should operate for all of the required symbol rates from 1 Ksps to 4 Msps, signal-to-noise ratios, and CIC-filter decimated sampling rates. It needs to track the carrier phase reliably when the received signal strength varies over many orders of magnitude.

The tracking loop bandwidth is programmable from 10 Hz to 10 KHz to meet the tracking and acquisition requirements for various communications scenarios. In addition, the carrier tracking loop needs to acquire and track received signals with maximum frequency offsets of +/- 10 KHz when the received signal is UHF and +/- 50 KHz when the received signal is at S-band. The tracking loop can also support navigation by supplying the instantaneous phase of the received signal.

Figure 7 shows the block diagram of MCAS1 carrier tracking loop. The loop follows the ADC, digital downconverter, and the CIC decimation filter (see Figure 4). The complex baseband loop input is multiplied by the complex output of the numerically-

controlled oscillator (NCO). The product of the complex multiplication is split into the real and the imaginary data paths. The signal path following the real output is termed the real arm of the carrier tracking loop and the path following the imaginary output is the imaginary arm of the loop.

SW 1 and SW 2 – Select residual carrier or Costas loop mode
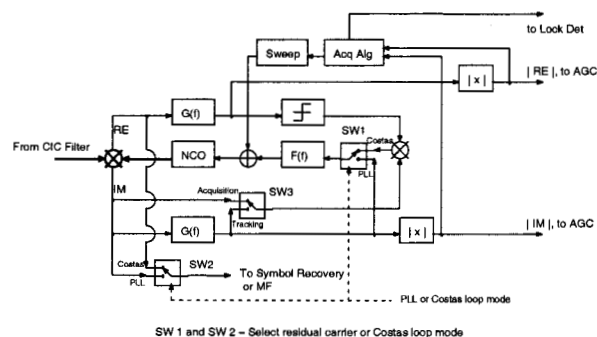SW 3 –Select acquisition or tracking mode for Costas loop

Figure 7. Block Diagram of the MCAS1 Carrier Tracking Loop.

Both the real and imaginary signals are filtered by a pair of identical low-pass arm filters, $G(f)$, with programmable cut-off frequency. The arm filters are discrete implementations of a first-order low-pass Butterworth filter. They are used to reduce noise in the carrier tracking loop; however, the arm filter cut-off frequency should not be so low that the output signal power is reduced significantly. It is found that the cut-off frequency which minimizes the tracking loop error for the arm filters is approximately equal to the received symbol rate, R_s, for non-return-to-zero (NRZ)-coded data.

When the received signal is a suppressed-carrier BPSK signal, the real arm filter output is passed through a hard limiter with output equal to +/- 1. It has been shown that with the operating Es/No at 0 dB or above, the limiter can reduce the squaring loss, S_L, of the Costas loops[4]. Squaring loss is caused by the multiplication of the real and imaginary arm signals. This operation is required to remove the data polarity. The penalty of this squaring operation is that noise in the imaginary arm is multiplied by both the signal and noise of the real arm resulting in poorer noise performance. With the hard limiter, the cross multiplier before the loop filter, $F(f)$, can be replaced by a combination of a switch and an inverter. The input to the loop filter is inverted when the output of the real arm filter output is negative. Instead of a real multiplier, the switch and inverter reduce the

complexity of the digital circuits, thus reducing the power consumption. A Costas loop with a hard limiter is termed a polarity-type Costas loop.

After the arm filters, one or both of the arm filter outputs are used to form the input to the loop filter, $F(f)$, depending on whether the tracking loop is operated in the suppressed carrier or residual carrier tracking mode (labeled PLL mode in Figure 7). There are three switches in the MCAS carrier tracking loop. SW1 and SW2 are selected depending on the suppressed or residual carrier operational mode. SW3 is used in the suppressed carrier mode. Its position is chosen depending on whether the tracking loop is in the acquisition or the tracking mode.

When the loop is operated in the residual carrier tracking mode, the imaginary part of the complex multiply product is used to drive the loop. For this mode of operation, Manchester data encoding is typically used to reduce the effects of data interference on residual loop tracking performance[1]. For either mode of operation (suppressed or residual carrier), tracking loop performance can be expressed in terms of loop SNR, ρ. For reliable tracking in the suppressed carrier mode, the loop SNR should be above the threshold: $\rho \geq 17dB$.

In the residual carrier operational mode, we impose the following minimum loop SNR guideline: $\rho \geq 14\ dB$, for good cycle slip and bit error rate (BER) characteristics.

We now briefly describe the acquisition algorithm for the MCAS1 carrier tracking loop in the suppressed carrier mode. The algorithm for the residual carrier tracking mode has not yet been developed, although it is expected to be very similar to that for the suppressed carrier mode which is described here. This algorithm is used to aid the digital carrier tracking (Costas) loop in acquiring phase/frequency lock. This is accomplished by sweeping through a user specified range (sweep range) of NCO frequencies at a user specified sweep rate and comparing the difference between the real and imaginary arm channel power estimates. The frequency sweeping is accomplished in discrete increments that are maintained for a user specified period of time. The real and imaginary arm power estimates are obtained by averaging M samples of data over each frequency increment interval. A lock detector output signal is generated from the difference between the real and imaginary arm filter output power estimates. This signal is then used to determine if the suppressed carrier tracking loop is locked by comparing it with a user programmable threshold. The

duration of each frequency increment and the lock detector threshold are functions of *Es/No* and the carrier tracking loop filter bandwidth.

The algorithm is always in one of two states: (i) in frequency/phase lock (verification state) or (ii) out of frequency/phase lock (acquisition state). The user may specify that N1 = 1, 2, 3 or 4 consecutive threshold "hits" by the lock detector output indicates that the carrier tracking loop is in phase and frequency lock before the algorithm transitions from acquisition to verification state. Likewise, the user may specify that N2 = 1, 2, 3 or 4 consecutive threshold "misses" indicates that the loop is not in phase and frequency lock before the algorithm transitions from a verification state back to an acquisition state. When N1 > 1, the probability of false lock ($P_{FL-Total}$) is given by[4]:

$$P_{FL-Total} = (P_{FL})^{N1} ,$$

where P_{FL} denotes the false lock probability when N1 = 1. Similarly, when N2 > 1, the probability of false alarm ($P_{FA-Total}$) is given by[4]:

$$P_{FA-Total} = (P_{FA})^{N2} ,$$

where P_{FA} denotes the false alarm when N2 = 1.

The algorithm allows the user to clear the loop filter registers after every frequency sweep. This is required when sweeping over large frequency ranges at low SNRs. When the carrier tracking loop is not in lock, the values in the filter accumulators average to zero over long periods of time. However, it is possible for a bias to become present in the digital accumulators in the loop filter even when it is not in lock. This bias may be temporary, an anomaly in the noise sequence, or it may due to a DC bias in the receiver system. In any case, when such a bias exists it can delay or even preclude loop acquisition. This problem can be eliminated by periodically clearing the filter accumulators when the algorithm is in the acquisition state. The maximum rate at which the carrier tracking loop filter registers can be cleared is equal to the rate that the NCO frequency is incremented, i.e., every M samples.

The Costas loop can false lock onto harmonics of the data that are half multiples of the data rate[5]. To counteract these false locks, the imaginary Costas arm filter can be switched out of the Costas loop (see Figure 7). This greatly reduces the probability of false lock[6]. For $E_s / N_0 \leq 20\,dB$, this virtually eliminates false locks. However, for $E_s / N_0 > 20\,dB$, false locks are still possible but

unlikely[5]. For data rates that are less than or equal to twice the sweep range, the algorithm switches the Q channel arm filter out of the Costas loop during the acquisition state and back into the Costas loop once the acquisition algorithm transitions to the verification state. The algorithm provides the real and imaginary arm filter output power estimates as well as the number of attempts to acquire and the no-lock/lock flag to the user through status registers.

4.2 NAVIGATION: DOPPLER PHASE MEASUREMENT

Missions like the Mars Relay Orbiter may be required to provide Doppler estimates derived from the received signal. This section describes how MCAS1 Doppler frequency estimates are obtained. The method described herein is applicable to either the suppressed or residual carrier tracking operational modes. The technique basically derives the Doppler frequency estimate from the difference between two instantaneous phase outputs from the phase register of the NCO. The resulting frequency estimate f_{est} is equivalent to counting the elapsed phase cycles over a fixed time interval. It is given by:

$$f_{est} \equiv \frac{\theta(t+T) - \theta(t)}{2\pi T} ,$$

where T is the time between two instantaneous phase measurements, $\theta(t)$ and $\theta(t+T)$. The standard deviation of the Doppler frequency estimation error is inversely proportional to T. Thus, T should be as large as possible while still yielding a meaningful frequency estimate. Typically, T is on the order of $10 - 60$ secs for a 10 minute pass.

In order to achieve 1 km accuracy for one-way Doppler positioning, the frequency measurements should have an accuracy equivalent to 1 mm/s for one-minute averaging using an oscillator with an Allan Deviation of 2×10^{-10} or less[1]. At 400 MHz, 1mm/s corresponds to 0.0013 Hz. Assuming conservatively that $T = 10\,sec$, 1 km navigational accuracy requires a carrier tracking loop SNR: $\rho \geq 24.6$ dB. This requirement for loop SNR is more stringent than that typically required for communications using a suppressed carrier tracking loop as discussed in Section 4.1, i.e., $\rho \geq 17$ dB. Thus, navigational requirements impose tighter constraints on demodulator design than do communication requirements.

4.3 SYMBOL TIMING RECOVERY

The symbol timing recovery algorithm is based on the absolute value type of early-late gate symbol synchronizer[7]. A combination of the early-late gate circuit and a random walk filter are employed as a symbol timing estimator as shown in Figure 8. This subsystem generates a timing signal that can be used to either jitter the symbol clock or the sample clock of the ADC. The nominal mode of operation calls for jittering the ADC clock because this minimizes the sampling offset induced degradation in BER performance. In certain situations where it is desirable to optimize navigation performance, the symbol clock may be jittered thus keeping a near constant time base for the phase samples entering the tracking loop.
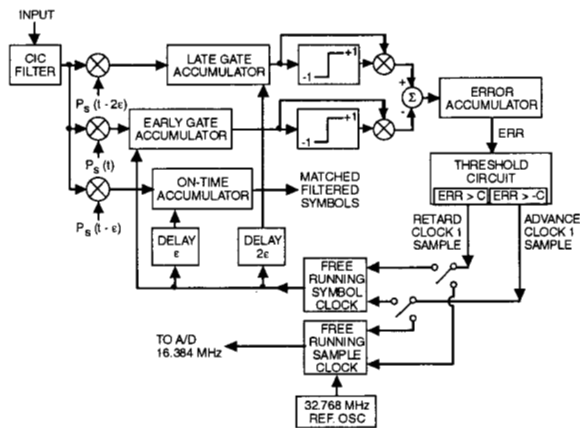

Figure 8. Block Diagram of the Symbol Timing Estimator.

The noisy baseband received symbols for which the timing is to be recovered are first routed though a CIC decimation filter as indicated in Figure 8. The CIC is identical to that defined in Section 3.3 (Figure 6) and is used to decimate to 16 samples/symbol for the lower symbol rates (i.e., 1,2,4, ..., 1024 ksym/sec) and thus narrow the bandwidth and improve the SNR prior to timing recovery. At 2048 ksym/sec, 8 samples/symbol are used for timing recovery and at 4096 ksym/sec, 4 samples/symbol are used.

The early-late gate symbol synchronization algorithm is motivated by the maximum a posteriori (MAP) estimation of an unknown parameter in Gaussian noise. As with the MAP estimator, the received BPSK signal that has passed through the CIC is first routed to cross correlators. The correlators operate on three different subintervals of the received signal (i.e., early, late and on-time), multiplying the received signal by an appropriately delayed stored replica of the transmitted pulse and then integrating (see Figure 8).

The nominal operating mode of the symbol timing recovery circuit corresponds to NRZ pulse shaping in which case the pulse shape depicted in Figure 8 is $P_s(t) = 1$. In the case of the Manchester encoded pulse shape, a stored replica of the Manchester $P_s(t)$ (a binary one is represented as a one for the first half of the symbol period and a zero for the second half of the symbol period) is multiplied by the received signal for each of the three arms of the timing recovery circuit.

The multiplications by the pulse shape are followed by an early integrator, an on-time integrator, and a late integrator. The three integrators are controlled by synchronous clocks that are slightly offset in time. The integration period is T_s for all three integrate and dump channels with the early integration starting a quarter of a symbol early and the late integration starting a quarter of a symbol late (i.e., $\varepsilon = T_s/4$ in Figure 8, which is optimal[7]). The on-time integration is equivalent to a matched filter for the transmitted symbols when the symbol timing is perfectly aligned.

The outputs from the early and late integrators are utilized to generate a timing error signal by differencing their respective absolute values. If the value of the early channel is greater that than of the late channel an error signal is generated that slows down the clock and if the late channel value is greater than that of the early channel a signal is generated to speed up the clock. The absolute value process eliminates the dependence of the error signal on bit polarity.

The timing error signal is used to advance/retard the phase of either the free running symbol clock or the free running sample clock. The timing jitter induced by noise is suppressed by a random walk filter[8]. The random walk filter consists of an error accumulator and threshold circuit. The accumulator is essentially an up/down counter accumulating the advance/retard errors (ERR in Figure 8). If ERR exceeds the threshold, C, then a correction pulse will be issued to correct for the jitter accordingly.

The resulting timing estimate, cl(t), is a stream of impulses nominally at the input symbol rate, 1/T, as depicted in Figure 9. We denote the error between the time-of-occurrence of the k-th impulse, τ_k, and the true, k-th symbol transition time, $t_k = t_0 + kT$, as:

$e_k = \tau_k - t_k$, which is termed the timing jitter error. The symbol timing recovery subsystem is required to

maintain an instantaneous timing jitter (e_k) satisfying:

$$e_k / T \leq 0.01.$$

The random walk filter has two modes of operation, acquisition and tracking. During acquisition the modulator transmits a pattern with guaranteed transitions every symbol. The random walk filter constant (C in Figure 8) will be set at a low value ($1 \leq C \leq 100$) during acquisition to allow rapid symbol synchronization. After synchronization is detected, the random walk filter is switched to a much higher value during the tracking mode to prevent symbol slippage ($100 \leq C \leq 10000$). The value chosen for C in the tracking mode is dependent upon the instantaneous jitter requirement given above ($e_k / T \leq 0.01$) as well as external factors such as oscillator stability. The actual operational value will be selected experimentally during development.
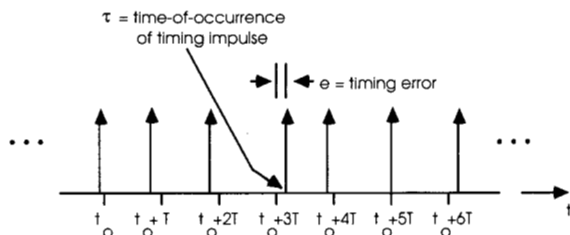

Figure 9. Symbol timing signal, cl(t).

4.4 CONVOLUTIONAL DECODER

The Viterbi decoder is utilized to provide error correction of the received symbols that were convolutionally encoded at the transmitter. The decoder utilizes soft decision symbol inputs to calculate the branch and state metrics of the trellis state transition diagram which are utilized to determine the maximally likely transmitted symbols. The Mentor Iventra Viterbi Encoder/Decoder soft core is used to implement the Viterbi decoder. It is a synthesizable Verilog RTL model[2].

The performance of this decoder has been evaluated using a bit-true C behavioral model. Based on these simulations, an E_b/N_o of 4.25 dB is required to achieve a BER of 10^{-5} when the input symbols are quantized to 5 bits.

The Mentor Viterbi decoder detects an out-of-sync condition but requires implementation of external logic to take the action to swap the input symbols upon indication of out-of-sync. Additionally, to be compatible with the CCSDS standard an inverter must be incorporated in the design external to the decoder as shown in Figure 1.

4.5 DIFFERENTIAL DECODER

The Viterbi decoder output can be passed to the differential decoder. The differential decoder can be enabled or disabled. The differential decoder performs the same differencing operation as the encoder to determine the original transmitted bit. It should be noted that differential encoder/decoder can cause error multiplication (i.e., one differential decoder input bit error can corrupt two output bits) resulting in a 0.2 dB loss when enabled.

4.6 DESCRAMBLER

The descrambler follows the differential decoder in the receive chain and can be enabled or disabled. The descrambler is discussed in Section 2. It should be noted that descrambling can cause error multiplication and incurs a 0.3 dB loss when enabled.

5. SUMMARY

This paper has provided a detailed overview of the MCAS1 communications system including the digital design considerations leading to a low mass and power transceiver. The system is being designed to operate over a wide range of data rates from 1 kbps to 4 Mbps and must accommodate frequency uncertainties up to 10 kHz with navigational Doppler tracking capabilities. As such, the design is highly programmable and incorporates efficient front-end digital decimation architectures to minimize power consumption requirements.

MCAS1 implements most of its transceiver functions with digital FPGA/ASIC technology. This leaves just the single stage RF up and down conversion to be implemented in the analog domain. The approach with the RF subsystem design is to use space-qualified parts where available and leverage the large investment that industry has made in developing highly integrated devices for the commercial wireless markets. The ultimate goal of the MCAS communications effort is to enable reliable communications at a significant mass, power, size and cost reduction for a broad class of very small platforms requiring short range communications.

REFERENCES

1. M. Agan, et.al., "MCAS1: A Low Power/Mass In Situ Transceiver," The Telecommunications and Mission Operations Progress Report 42-138, April-June 1999, Jet Propulsion Laboratory, Pasadena, California, August 15, 1999, in preparation.

2. Viterbi Encoder/Decoder Technical Product Brief, Mentor Graphics Corp., 1997.

3. Hogenauer, "An economical class of digital filters for decimation and interpolation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, v. 34, April 1981, pp. 155-162.

4. *Deep Space Telecommunications Systems Engineering*, J. Yuen (Ed.), Plenum Press, NY, 1983.

5. M. Simon, "The False Lock Performance of Costas Loops with Hard-Limited In-Phase Channel", IEEE Transactions On Communications, VOL. VOM-26, NO. 1, January 1978.

6. C. Cahn, "Improving Frequency Acquisition of a Costas Loop", IEEE Transactions On Communications, VOL. COM-25, NO. 12, December 1977.

7. W. Lindsey and M. Simon, *Telecommunication Systems Engineering*, Prentice Hall, Englewood Cliffs, New Jersey, 1973.

8. D. Paranchych, N. Beaulieu, "Use of Second Order Markov Chains to Model Digital Symbol Synchroniser Performance", IEE Proc.-Commun., VOL. 143, NO. 5, October 1996.